A Neuromorphic Computational Primitive for Robust Context-Dependent Decision Making and Context-Dependent Stochastic Computation

Liang, Dongchen ; Indiveri, Giacomo

Abstract: The prefrontal cortex (PFC) plays an important role in complex cognitive computations, including planning and decision making. Although recurrent spiking neural network (SNN) software models of PFC have been successful in reproducing many of its cognitive computational aspects, little attention has been devoted to the question of how such systems can perform with low-resolution parameters and be robust to noise and variability in their input signals and state variables. Here, we present a mixed-signal analog/digital neuromorphic implementation of a state-dependent SNN architecture that addresses these issues by construction. The network relies on synaptic dis-inhibition to ensure robust decision making even in the face of very large variability. Depending on its connectivity, the network can either perform robustly in a deterministic way or exploit the device mismatch and noise to explore stochastically multiple states in constraint satisfaction problems (CSPs). We validate the architecture by mapping it onto a network of spiking neurons in a multi-core mixed-signal neuromorphic system and presenting experimental results for three different examples of CSPs.

DOI: https://doi.org/10.1109/tcsii.2019.2907848

A Neuromorphic Computational Primitive for Robust Context-Dependent Decision Making and Context-Dependent Stochastic Computation

Dongchen Liang, *Member, IEEE* and Giacomo Indiveri, *Senior Member, IEEE*

***Abstract*—The prefrontal cortex (PFC) plays an important role in complex cognitive computations including planning and decision making. Although recurrent spiking neural network (SNN) software models of PFC have been successful in reproducing many of its cognitive computational aspects, little attention has been devoted to the question of how such systems can perform with low-resolution parameters and be robust to noise and variability in their input signals and state variables. Here we present a mixed-signal analog/digital neuromorphic implementation of a state-dependent SNN architecture that addresses these issues by construction. The network relies on synaptic dis-inhibition to ensure robust decision making even in the face of very large variability. Depending on its connectivity, the network can either perform robustly in a deterministic way or exploit the device mismatch and noise to explore stochastically multiple states in Constraint Satisfaction Problems (CSPs). We validate the architecture by mapping it onto a network of spiking neurons in a multi-core mixed-signal neuromorphic system and presenting experimental results for three different examples of CSPs.**

***Index Terms*—Mixed-signal neuromorphic system, noisy spiking neural network, Constraint Satisfaction Problems, Winner-Take-All, Neural State Machines.**

I. Introduction

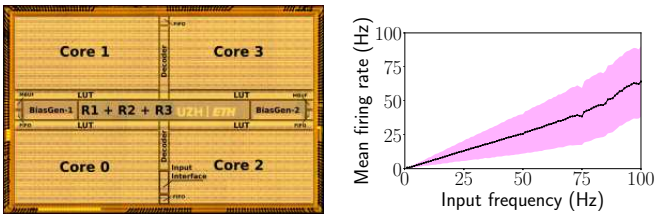
IT is believed that the prefrontal cortex (PFC) constructs an internal representation of the world [1] to detect conflicts among multiple representations [2] and to coordinate different brain areas for reducing the cost of interference and confusion [3], thus establishing a stable mapping between the input, internal representations, and the output [4], [5]. However, the specific network architecture through which neurons perceive the conflicts and reach the maximum consistency solution is still unknown. Several software and hardware models have been proposed to explain how the conflicts are avoided and how the stable state is approached, using attractor dynamics and competitive Winner-Take-All (WTA) networks [6], [7], [8], [9], [10], [11], [12], [13]. However, most of these models are based on the assumption that the network can get out of local minima thanks to external stochastic stimuli or injected noise currents. In this work, we address the question of how a spiking neural network (SNN) can smoothly shift its mode of operation from a noisy-stochastic substrate to a reliable one to achieve the maximum consistency state, without having to resort to external noise sources. Here we present a SNN model that addresses this question.

Biological neural networks perform state-dependent computations with working memories [5], [14], [15]. Many studies have shown that working memory is maintained by recurrent excitation and modulated by feedback inhibition [16], [17]. In the cortical layer 2/3, which has the highest percentage of recurrent excitations in cortical layers, WTA structures are widely observed [18]. They have been considered as a primitive structure and canonical microcircuit of the cortex [18], and have been proposed as fundamental computational primitives for synthesizing spike-based cognitive systems [19]. Modeling studies on state-dependent computations [19], [20] have shown that the same signal can trigger the transition between the winners of coupled WTA networks. However, when designing the network architectures, these studies rarely take into account the fact that hardware (both biological and electronic) implementations of such SNNs are affected by variability in the state parameters and have distributions of values, rather than bit precise parameter values. Similar to what is observed in biology [21], analog circuits are affected by the variance in their physical substrate and therefore have an amount of inherent mismatch [22], [23]. The network scheme we propose uses dis-inhibition to increase the robustness to parameter variation and uses both short-range and long-range connections to enable robust coordination among distributed local WTA clusters. We exploit the device mismatch in silicon neurons and synapses to produce randomized neural activity in the time domain within the recurrent network (as also demonstrated in [24]), allowing the network to perform stochastic computations without having to inject externally supplied noise explicitly. We call the proposed architecture a *Cortical Automaton (CA)*, and we demonstrate its effectiveness as a powerful computational primitive by implementing it on a recently developed mixed-signal neuromorphic processor (the Dynamic Neuromorphic Asynchronous Processor (DYNAP)) [22], and by applying it to the solution of a variety of Constraint Satisfaction Problems (CSPs), which represent a class of decision-making problems. Unlike noise-driven CSP solvers, we show how the CA can explore the state-space escaping local minima, and staying fixed in the global minimum once it finds it (a significant improvement over [11]).

In the next section we first give a brief summary of the characteristics of the DYNAP device used to implement our architecture; in Section III we describe the CA architecture itself; Section IV describes the CSPs solved and Section V

This work is supported by the China Scholarship Council (CSC) and by the Institute of Neuroinformatics, University of Zurich and ETH Zurich.

The authors are with the Institute of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland. (e-mail: dongchen@ini.uzh.ch; giacomo@ini.uzh.ch).



(a) (b)

Fig. 1. DYNAP neuromorphic chip. (a) The layout. In each chip, there are 4 cores. In each core, there are 256 silicon neurons. Neurons communicate with each other through a 3-stage routing system, namely the routers R1, R2, and R3. They are responsible for the communication between neurons in the same core, in the same chip but different cores, and in different chips respectively. Two 'BiasGen' modules are used for tuning the parameters of the silicon neurons, such as threshold, time constant, synapse weight, etc. (b) Variation of the firing rate of 256 neurons sharing the same parameters in function of the frequency of input stimuli. The shaded area shows the standard deviation.

presents the experimental results.

II. Mixed-Signal Neural Processing Substrate for Emulating Spiking Neural Networks

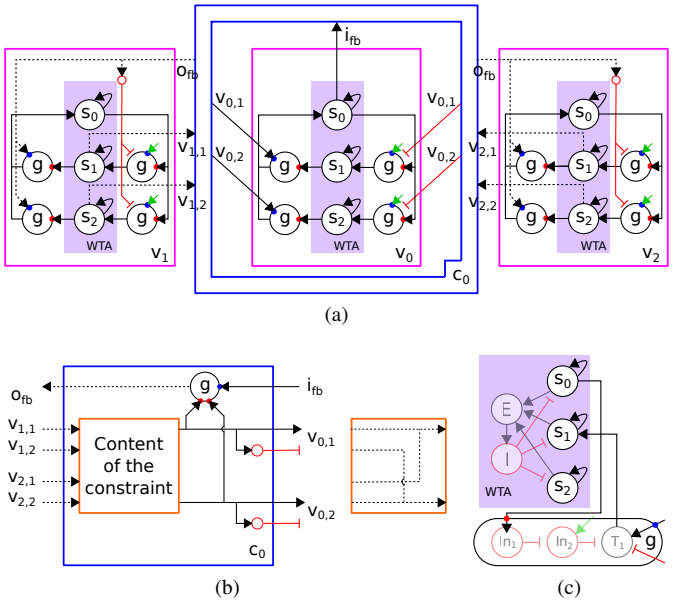
The DYNAP device used in our setup comprises four chips arranged in a 2×2 array. Each chip has four cores of 256 neurons each. Each neuron has 64 synaptic inputs that can be programmed to accept Address-Events from any other neuron of any of the four chips and can transmit its output spike to the synapses of all 1024 neurons, of up to four different chips (thus with a maximum fan out of 4096). Neuron and synapse circuits are analog, with biologically plausible dynamics [25], while network routing and mapping circuits are asynchronous digital, using a memory-optimized hierarchical routing scheme [22]. Figure 1b illustrates a measurement of the inherent variance among neurons. The coefficient of variation (CV) in the measurements can vary between 8% and 20% [26].

III. Spiking Neural Network Architecture

The network architecture of a CA is illustrated in Fig. 2. It is composed of two main sub-networks: variable-encoding networks (see $v_0, v_1 \dots v_n$ of Fig. 2) and constraint networks (see $c_0, c_1 \dots c_n$). Each *variable-encoding network* represents a variable of the CSPs. The variable value is represented by a winning population of neurons in a WTA network. WTA winners can also represent meta-stable states, which denote that the variable is currently not assigned a value. The *constraint networks* define the relationship (constraint) between variables.

A. Variable-encoding network

The variable-encoding network is implemented by coupled WTA networks arranged to form the Neural State Machine (NSM) architecture described in [27], where a group of "gate units" (denoted as g in Fig. 2) is used to gate the WTA winner node transitions through a dis-inhibition mechanism. The "gate units" realize robust symbolic AND behaviors with 100% success rate in the face of the large variability in the chips. This enables robust state-dependent computations using a small number of neurons per population, as explained



(a) (b) (c)

Fig. 2. The primitive structure of variables and constraints. Each circle denotes a population of spiking neurons. The solid lines denote fixed synapses, while the dotted lines denote synapses that can be customized for different CSPs. The black and red lines denote excitatory and inhibitory synapses respectively. The green arrows denote constant stimuli. (a) shows the connectivity between variables and constraints and the internal structure of variables. The internal structure of a constraint (the blue box of (a), denoted as c_0) is shown in (b). The content of the constraint can be expressed as the connectivity within the orange box. An example of the connectivity is illustrated on the right of (b): Neighboring variables should not share the same value (to solve a Sudoku task). The "gate units" g, would fire only if all the inputs (denoted as red and blue dots) are both supplied. (c) The "gate units" are implemented by a three-node dis-inhibitory network (see also [27]). The constantly firing neurons E supplies the constant stimuli for each v_i and the linked c_i.

by [27]. The stochastic search process is based on the interactions (affected by the inherent stochasticity) among local robust state-dependent computation in the variable-encoding networks. In the example of Fig. 2a, each variable has two potential values, e.g., 0 and 1. They are represented by the states s_1 and s_2 respectively. The idle state, s_0, indicates that currently this variable is not assigned a value: The variable has to abandon the current value and select a new one.

B. Constraint network

Figure 2b illustrates the detailed structure of a constraint network. The content of a constraint is defined by its internal connectivity (see the orange box of Fig. 2b). An example of the connectivity is shown in the right of Fig. 2b. It is used to describe the constraint that the variable-encoding network v_0 should not share the same value with v_1 and v_2. The internal connectivity of c_0 determines which state of the other variable-encoding networks v_1 and v_2 are allowed to select given the current state of the variable-encoding network v_0. In CSPs, the constraints are always equally applied to all the connected variables. In the example of Fig. 2a, from any variable v_0, v_1, or v_2, the network should be able to infer the value of the others. To fully describe this bi-directional constraint, each of the variable-encoding networks v_1 and v_2 should also be linked to a constraint network (e.g., c_1 and c_2) and v_0 will be connected to them with the same principle as shown in

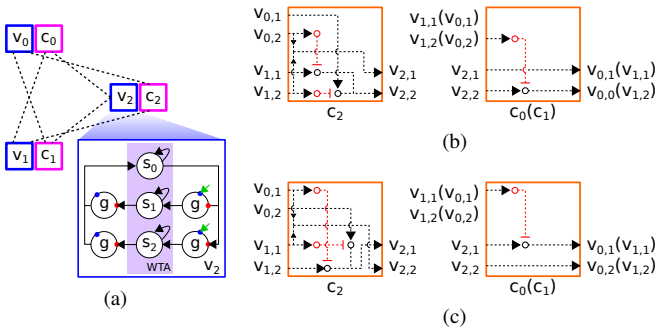


Fig. 3. A unit structure of the SAT problem. (a) shows a diagram of the connectivity between three variables: v_0, v_1, and v_2. Every variable is linked to a constraint. The dotted lines represent the connectivity illustrated in Fig. 2a. For each variable, the optional values True and False are represented by the states s_1 and s_2 respectively. For simplicity, the interconnections between variables and constraints are not illustrated. (b) and (c) illustrate the content of the constraint boxes. (b) To describe the AND relation: $v_2 = v_0 \wedge v_1$. (c) To describe the OR relation: $v_2 = v_0 \vee v_1$. In (b) and (c), each small circle represents a population of neurons. Black circles denote excitatory neurons, and red circles denote inhibitory neurons.

Fig. 2b. Figure 3 shows examples that implement the logic relations AND and OR in the CA architecture, where each variable-encoding network is linked to a constraint network. The NOT relation can be expressed using the same principle shown in Fig. 2b.

C. Two-phase mixed searching process

All the variable and constraint networks operate in parallel. They collaboratively determine the final equilibrium state of the network. The state-dependent computation ensures that any time the variables are at one of the potential states, either the idle state or the ones represent values. If the assigned values of two neighboring variables violate a constraint (e.g., in Fig. 2a, v_0 is at s_1, and v_1 is at s_1 as well), they will drive each other to go to the idle state s_0 (through the excitatory $v_{0,1}$ and $v_{0,2}$ signals in Fig. 2a). Due to the concurrent and stochastic nature of the searching process, in most cases, only one of the two variables will win the competition and stay at the original one, and the other one will end in the idle state. The winner variable will send inhibitory signals to the prevent the loser from selecting the same value again (through the inhibitory $v_{0,1}$ and $v_{0,2}$ signals in Fig. 2a). If a variable is in its idle state, the WTA will be driven to change state to one of the potential "value" states by constant stimuli. However, if all the "value" states are blocked by the inhibition, the variable will try to reset the neighboring variable-encoding networks to the idle state with the feedback connections i_{fb} and o_{fb}. This ensures that the variable will not hang in the meta-stable state s_0, and get stuck in the searching process. Once some of the neighbors are reset, and the inhibition is removed, the variable will select a "value" state. The same process will be repeated for its neighbors and all the other variables in parallel until a globally consistent solution is found. Thus the search process is not entirely random. It will follow the gradient (implemented by $v_{0,1}, v_{0,2}, \ldots$) to minimize the number of violated constraints (greedy phase), but once it detects a local minimum, it resets the local variables and triggers a new exploration (exploratory phase). Thus the search process is a mixture of greedy and exploratory phases across the local networks. This two-phase mixed searching process keeps running until it finds the optimal solution.

D. Stochasticity in a deterministic architecture

Different from previous studies, the search process in a CA does not require externally supplied noise. The stochasticity inherently exists in the mixed-signal hardware, due to the analog nature of the circuits and the device mismatch effects. The state neurons receive negative feedback (denoted as n) through the E and I neurons of the WTA structure and receive positive feedback (denoted as p) through recurrently connected excitatory synapses so that they can self-sustain their firing activities. The synaptic integrator circuits of the two feedbacks are configured to have different time constant τ ($\tau_n < \tau_p$) and synapse weight ω ($\omega_n > \omega_p$). Once the state neurons start to fire, the negative feedback first turns off their activity, and then the positive feedback activates them again. As this process happens repeatedly, the feedback leads to an intrinsic oscillatory behavior (e.g., see the raster plot experimental data of Fig. 6). Due to the inherent variance between neurons and synapses, any time receiving the same amount of spikes, different neurons in the same population regenerate a variant amount of new spikes to the others and to themselves, which results in the fluctuation of the oscillation frequency along with time. Effectively all the neurons in one population share the variance through interactions. In this way, the oscillating activities transfer the spatially distributed variability among neurons into the time domain: asynchronous oscillations among populations. The connectivity between variable-encoding networks imposes the influence of the asynchronous activities to each other. Thus the oscillating activity of each variable depends on not only its own feedbacks but also the feedback from its neighbors. The asynchronous oscillations ensure that there is no phase locking among the oscillatory neurons and no infinite loop during the search process. Also, the small thermal/jitter noise of circuits will spontaneously change the timing of spikes, and this small disturbance will be broadcast and amplified to the rest of this network through the recurrent connectivity.

IV. Constraint Satisfaction Problems (CSPs)

Here we describe the three tasks used to demonstrate the CSP solving abilities of the CA network.

A. Sudoku

In this task, all the cells are restricted by the constraints that any two cells in a row, in a column, or in a quadrant should not share the same value. Each variable is represented by a variable-encoding network. The potential values are denoted as the states of the variable-encoding network, as illustrated in Fig. 4. If the network were to select two variables with equal values, it would quickly destabilize the working memories and push the variables to change their value, until the solution to the Sudoku problem is found. The strategy to define the constraint that two variables should not share the value is the same one as discussed in Section III.

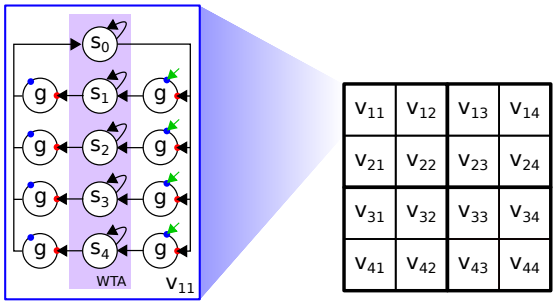


Fig. 4. 4×4 Sudoku task. Each cell has four potential values. Any two cells in a row, in a column, or in a quadrant should not share the same value. Each cell is implemented by a variable-encoding network that has five states. Four of them represent the four potential values of the cell.

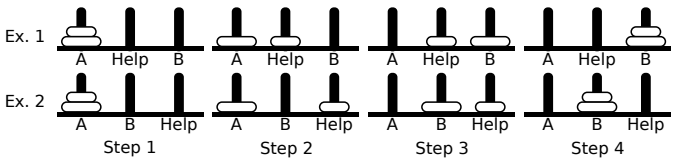


Fig. 5. Two examples of the Tower of Hanoi task. Each of them has three stacks and two disks. From left to right, the initial state, intermediate steps, and the final state are illustrated. The stack A and B denote the source and destination stacks respectively. The stack Help denotes an auxiliary stack that is necessary for finishing the moving of disks from source to destination.

B. Boolean Satisfiability Problem (SAT)

This problem is composed of variables, operators AND (conjunction, denoted by $\wedge$), OR (disjunction, $\vee$), NOT (negation, $\neg$), and parentheses. The SATs can be expressed in a conjunctive normal form, which is made up of the conjunction of a group of clauses (the disjunction of variables or the negation of variables). We test the proposed network model in a simple 3-SAT problem with three variables and two clauses: $(A \vee B \vee \neg C) \wedge (B \vee C \vee D) = T$. In this task, four variable-encoding networks are needed to represent the variables, and three additional ones are needed to represent the two clauses and the conjunction of them.

C. Tower of Hanoi

The Tower of Hanoi is a typical test for executive functions in cognitive neuroscience [28], [29]. The objective of this task is to move disks from one stack to another obeying the following rules: Only one disk can be moved at a time; Each move consists of taking the top disk from one stack and placing it on the top of another one; No disk can be placed on the top of a smaller one. Given these constraints, this task can be mapped into the form of CSPs. We validate it in a simple task using three stacks and two disks, as shown in Fig. 5. In detail, at each step, on which stack a disk is located is represented by a variable. The states of this variable denote the three stacks. Another group of variables is used to represent on which layer the smaller disk is located: at the bottom or above the larger one. Some additional variables are used to represent whether the disks are moved or not between two steps. The constraints can be expressed as the AND or OR relations between these variables and implemented in the same way as the SAT problem (see Fig. 3).

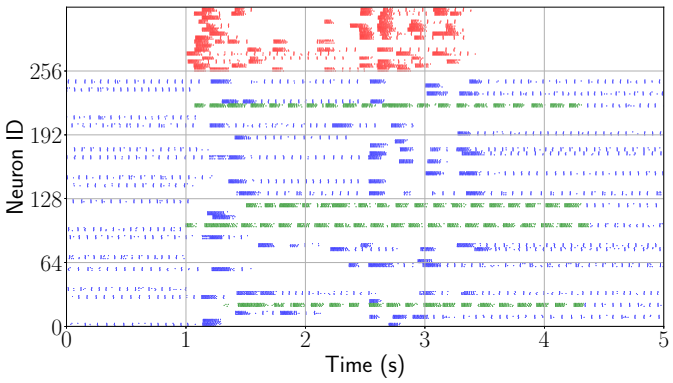


Fig. 6. Activity of the state neurons in a Sudoku task. Red: Neurons of the idle state s_0. Blue: Neurons of the "value" states s_1, s_2, s_3, and s_4. They represent the optional values of each cell. Initially, the network is at a solution (represented by the neurons of the "value" states). Between 1 s and 4.3 s, a 5 Hz external stimuli is given to the network to force four cells to alter their value as a given context (denoted as green). From 1 s to 3.5 s, the network finds a new solution. After the input removal, the network will not lose the found solution until a new trial starts. The state neuron groups are firing with a oscillatory (bursting) behavior around 15 Hz.

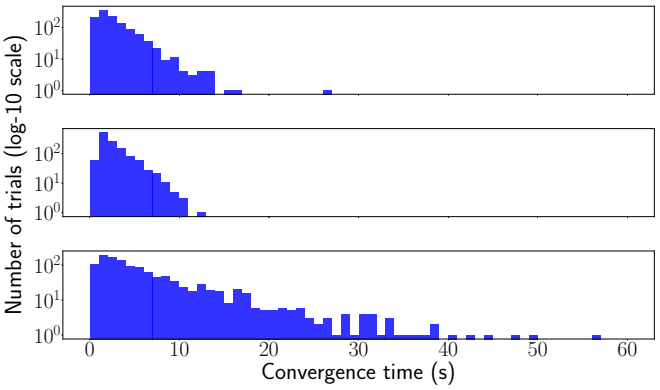


Fig. 7. Time cost by solving CSPs. We plot the histogram on a log-10 scale. There are 1200 random trials for each task. Top: The 4×4 Sudoku task. In each trial, there is a randomly initialized cell. Middle: The 3-SAT task of Section IV. Bottom: The Tower of Hanoi task illustrated in Fig. 5. On average, it would take 2.75 s, 2.65 s, and 6.36 s to solve each task respectively. The average cost is faster or comparable to the human performance [30], [31] (normalized by the total number of constraints).

V. Experimental Results

We mapped the architectures determined to solve the three CSPs described in Section IV onto the hardware setup comprising the DYNAP chips and measured the silicon neuron neural activity in response to the inputs provided. With the current hardware setup, we used 4 neurons for each neural population in the Sudoku and SAT task and 2 neurons in the Tower of Hanoi task. In total, we used 1920 neurons and 54748 synapses for the 4×4 Sudoku task, 1232 neurons and 8228 synapses for the 3-SAT task, and 1176 neurons and 16784 synapses for the Tower of Hanoi task.

1) Performance and robustness: The raster plot of Fig. 6 shows the neural activities when solving a 4×4 Sudoku task. When all the s_0 neurons stop firing, it denotes a solution has been found. Once a solution is found, the network will keep it unless new stimuli are given to start a new trial. In Fig. 6, we show a spike train with low frequency (5 Hz) is enough

to set a condition for the network to find a new solution. The network is robust to the input with high frequency ($>$1000 Hz). Thus, our model can compute with a wide range of input spike frequency. Figure 7 shows the speed of convergence in the 4×4 Sudoku, 3-SAT, and Tower of Hanoi tasks.

2) Power consumption: The mixed-signal asynchronous circuits work with ultra-low power consumption. All analog neuron and synapse circuits are current-mode, with no active amplifiers or components. Therefore, if there are no incoming events, the SNN architecture implemented on these neuromorphic devices only consumes static power (e.g., due to leakage currents). The static power dissipation is 945 μW for the DYNAP chip [32]. The main source of the dynamic power consumption is due to neurons firing and spikes generation. For generating every spike, the neurons use 2.8 pJ [32]. For example, in the Sudoku task, the average firing rate of the 1920 used silicon neurons is 25.06 Hz during the search phase and 16.65 Hz after they converge to an optimal solution. Even with conservative measures, it can be estimated that the total static plus dynamic power consumption of the system during this task will be safely below 1 mW.

VI. Conclusion

We proposed a SNN architecture to coordinate the computation of local WTA. We showed how this deterministic architecture can be configured to perform both robust state-dependent computations and stochastic search without requiring external sources of noise. We showed how its basic elements can be combined to synthesize networks that solve CSPs robustly and efficiently. We validated the architecture with mixed-signal analog/digital neuromorphic hardware experiments. This work might shed light on the computational benefits of WTA structures in biological neural networks.

Acknowledgment

This work is supported by the China Scholarship Council (CSC) and by the Institute of Neuroinformatics, University of Zurich and ETH Zurich.

References

[1] E. K. Miller and W. F. Asaad, "The prefrontal cortex: conjunction and cognition," *Handbook of Neuropsychology*, vol. 7, pp. 29-54, 2002.

[2] M. M. Botvinick, T. S. Braver, D. M. Barch, C. S. Carter, and J. D. Cohen, "Conflict monitoring and cognitive control," *Psychological Review*, vol. 108, no. 3, p. 624, 2001.

[3] E. Miller and J. Cohen, "An integrative theory of prefrontal cortex function," *Annual Review of Neuroscience*, vol. 24, no. 1, pp. 167–202, 2001.

[4] M. G. Stokes, M. Kusunoki, N. Sigala, H. Nili, D. Gaffan, and J. Duncan, "Dynamic coding for cognitive control in prefrontal cortex," *Neuron*, vol. 78, no. 2, pp. 364–375, 2013.

[5] K. D. Harris and A. Thiele, "Cortical state and attention," *Nature Reviews Neuroscience*, vol. 12, no. 9, p. 509, 2011.

[6] X.-J. Wang, "Neural dynamics and circuit mechanisms of decision-making," *Current Opinion in Neurobiology*, vol. 22, no. 6, pp. 1039–1046, 2012.

[7] M. Pfeiffer, B. Nessler, R. J. Douglas, and W. Maass, "Reward-modulated hebbian learning of decision making," *Neural Computation*, vol. 22, no. 6, pp. 1399–1444, 2010.

[8] F. Corradi, H. You, M. Giulioni, and G. Indiveri, "Decision making and perceptual bistability in spike-based neuromorphic VLSI systems," *2015 International Symposium on Circuits and Systems (ISCAS)*. IEEE, pp. 2708–2711, 2015.

[9] H. Mostafa, L. K. Mller, and G. Indiveri, "Rhythmic inhibition allows neural networks to search for maximally consistent states," *Neural Computation*, vol. 27, pp. 2510–2547, 2015.

[10] Z. Jonke, S. Habenschuss, and W. Maass, "Solving constraint satisfaction problems with networks of spiking neurons," *Frontiers in Neuroscience*, vol. 10, p. 118, 2016.

[11] J. Binas, G. Indiveri, and M. Pfeiffer, "Spiking analog VLSI neuron assemblies as constraint satisfaction problem solvers," *2016 International Symposium on Circuits and Systems (ISCAS)*. IEEE, pp. 2094–2097, 2016.

[12] G. A. Fonseca Guerra and S. B. Furber, "Using stochastic spiking neural networks on spinnaker to solve constraint satisfaction problems," *Frontiers in Neuroscience*, vol. 11, p. 714, 2017.

[13] U. Rutishauser, J.-J. Slotine, and R. J. Douglas, "Solving constraint-satisfaction problems with distributed neocortical-like neuronal networks," *Neural Computation*, vol. 30, no. 5, pp. 1359–1393, 2018.

[14] T. L. Cheng-yu, M.-m. Poo, and Y. Dan, "Burst spiking of a single cortical neuron modifies global brain state," *Science*, vol. 324, no. 5927, pp. 643–646, 2009.

[15] M. L. Schlvinck, A. B. Saleem, A. Benucci, K. D. Harris, and M. Carandini, "Cortical state determines global variability and correlations in visual cortex," *Journal of Neuroscience*, vol. 35, no. 1, pp. 170–178, 2015.

[16] R. Douglas, C. Koch, M. Mahowald, K. Martin, and H. Suarez, "Recurrent excitation in neocortical circuits," *Science*, vol. 269, pp. 981–985, 1995.

[17] A. Compte, N. Brunel, P. S. Goldman-Rakic, and X.-J. Wang, "Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model," *Cerebral Cortex*, vol. 10, no. 9, pp. 910–923, 2000.

[18] R. Douglas and K. Martin, "Recurrent neuronal circuits in the neocortex," *Current Biology*, vol. 17, no. 13, pp. R496–R500, 2007.

[19] E. Neftci, J. Binas, U. Rutishauser, E. Chicca, G. Indiveri, and R. Douglas, "Synthesizing cognition in neuromorphic electronic systems," *Proceedings of the National Academy of Sciences*, vol. 110, no. 37, pp. E3468–E3476, 2013.

[20] U. Rutishauser and R. Douglas, "State-dependent computation using coupled recurrent networks," *Neural Computation*, vol. 21, pp. 478–509, 2009.

[21] A. A. Faisal, L. P. Selen, and D. M. Wolpert, "Noise in the nervous system," *Nature Reviews Neuroscience*, vol. 9, no. 4, pp. 292–303, 2008.

[22] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs)," *IEEE transactions on Biomedical Circuits and Systems*, vol. 12, no. 1, pp. 106–122, 2018.

[23] "Brain-inspired multiscale computation in neuromorphic hybrid systems (BrainScaleS)," *FP7 269921 EU Grant*, 2011–2015.

[24] E. Chicca and S. Fusi, "Stochastic synaptic plasticity in deterministic aVLSI networks of spiking neurons," *Proceedings of the World Congress on Neuroinformatics*, pp. 468–477, 2001.

[25] E. Chicca, F. Stefanini, C. Bartolozzi, and G. Indiveri, "Neuromorphic electronic circuits for building autonomous cognitive systems," *Proceedings of the IEEE*, vol. 102, no. 9, pp. 1367–1388, Sep. 2014.

[26] N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, and G. Indiveri, "A re-configurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses," *Frontiers in Neuroscience*, vol. 9, no. 141, pp. 1–17, 2015.

[27] D. Liang and G. Indiveri, "Robust state-dependent computation in neuromorphic electronic systems," *2017 Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, pp. 108–111, 2017.

[28] S. R. Beers, D. R. Rosenberg, E. L. Dick, T. Williams, K. M. OHearn, B. Birmaher, and C. M. Ryan, "Neuropsychological study of frontal lobe function in psychotropic-naive children with obsessive-compulsive disorder," *American Journal of Psychiatry*, vol. 156, no. 5, pp. 777–779, 1999.

[29] N. A. Zook, D. B. Davalos, E. L. DeLosh, and H. P. Davis, "Working memory, inhibition, and fluid intelligence as predictors of performance on tower of hanoi and london tasks," *Brain and Cognition*, vol. 56, no. 3, pp. 286–292, 2004.

[30] R. Pelnek, "Difficulty rating of sudoku puzzles by a computational model," *Twenty-Fourth International FLAIRS Conference*, pp. 434–439, 2011.

[31] J. R. Anderson and S. Douglass, "Tower of hanoi: Evidence for the cost of goal retrieval," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 27, no. 6, p. 1331, 2001.

[32] G. Indiveri, F. Corradi, and N. Qiao, "Neuromorphic architectures for spiking deep neural networks," *2015 IEEE International Electron Devices Meeting (IEDM)*. IEEE, pp. 4.2.1–4.2.14, 2015.